BOOK FIVE · THE AI ECONOMY MONETIZATION SERIES

Monetizing Service as Software

in the AI Economy

The question is not whether AI has changed what you sell. It has. The question is whether your pricing model knows it yet.

When AI agents do the work, you are no longer selling software. You are selling a service delivered at software scale. The pricing, contracting, and economics of that category are fundamentally different from both traditional software and traditional services.

Framework F19: The Service-Software Convergence Model

PREFACE

When the Product Stops Being a Tool

The moment of category shift — why it arrives gradually, then all at once, and why it changes everything.

There is a moment in the history of every AI software company when the product stops being a tool and becomes a service. The moment is rarely announced. It arrives gradually, then all at once.

It arrives the first time a customer says: "We don't use the product — the AI uses the product for us." It arrives when the customer success conversation shifts from "are your users adopting it?" to "is it working?" It arrives when the renewal conversation is no

longer about how many seats the customer needs but about what the AI accomplished in the past year.

This moment represents a fundamental category shift. The company was selling software — a tool that customers operated to produce results. Now it is selling a service — results delivered by an AI operating the tool on the customer's behalf. The customer's experience of value has changed. Their relationship to the product has changed. Their commercial expectations have changed.

What has not changed — in most companies — is the pricing model. The company is still charging per seat, or per token, or per API call. It is still billing for access to the tool rather than for the value the tool delivers. The commercial architecture is stuck in the old category while the product has moved into a new one.

This mismatch between what the product is and how it is priced is the central commercial challenge for software companies in the AI era. It is what this book is about.

The book has four parts. Part One examines the convergence: what happened historically, what companies are navigating it right now, and why the existing commercial models — both traditional software and traditional services — are inadequate for the new category. Part Two develops five pricing models specifically designed for Service as Software, each with worked examples, implementation guidance, and a clear analysis of when each model fits and when it fails. Part Three covers the operational infrastructure: contracting, revenue recognition, and the delivery mechanisms that make outcome-based commitments operationally real. Part Four examines the strategic implications: how to defend margin in this new category, how customer success works when the product runs autonomously, and how to govern AI service delivery at enterprise scale.

The argument throughout is that Service as Software is not a variant of SaaS. It is a new category with its own commercial logic, its own accounting treatment, and its own relationship between vendor and customer. The companies that understand this and design for it will build extraordinarily durable businesses. The ones that treat it as

upgraded SaaS will find that their pricing models cannot capture the value they are creating.

PART ONE

The New Category

What happened. Who is navigating it. Why existing models fail at the convergence point.

CHAPTER ONE

The Convergence: When Software Became a Service

The historical moment. Salesforce Agentforce, Harvey AI, Cognition Devin. What changed and why.

In January 2023, a small law firm in Austin, Texas began using an AI system to review contract drafts before they went to partner review. The system was sold to them as legal AI software — a tool that their lawyers would use to check their work. The firm paid a monthly subscription for access.

Six months later, the firm had restructured its associate workflow. When a client sent a contract for review, the AI reviewed it first, flagged the significant issues with draft language suggesting alternatives, estimated the risk level, and prepared a summary for the partner. The associates' job had shifted from reviewing contracts to reviewing the AI's reviews. The AI was doing, by volume of work, about 70% of what the associates had previously done.

The firm's CFO asked a question at their annual software review that the vendor had not been asked before: "We're paying you a monthly subscription for legal AI software. But

what your AI is actually doing is contract review work. Can you tell us what that's worth?"

The vendor had no answer. They had designed their pricing for the tool, not for the work the tool was doing. The CFO was asking about the work.

This gap — between the value delivered (contract review work, at scale, continuously) and the commercial model designed to capture it (a monthly software subscription) — is the open-claw problem at its most visible. The AI had become a service provider. The commercial model had not caught up.

The same dynamic is playing out across every sector where AI is deployed at operational depth. The specific companies involved, the specific workflows being transformed, and the specific pricing gaps that emerge differ — but the structural pattern is identical: a software product that was sold as a tool to be operated by humans has become a service delivered by AI on behalf of humans. The customers understand this shift. The vendors often do not.

> ***"The customer is no longer buying a tool. They are buying the work. That single change rewrites the commercial logic of the entire category."***

Three Companies Navigating the Convergence

Three specific companies have navigated the convergence in ways that illuminate both the opportunity and the difficulty.

Salesforce's journey with Einstein and Agentforce is the largest-scale example of a software company navigating the service-software convergence. For twenty years, Salesforce sold CRM software — tools that salespeople used to manage their pipelines,

log their activities, and forecast their numbers. Salespeople were the users. Salesforce charged per seat.

Einstein AI, launched in 2016, began the transformation: AI recommendations embedded in the CRM, helping salespeople make better decisions. Still a tool. Still per seat. Then Agentforce, launched in 2024, completed the shift: autonomous AI agents that could qualify leads, respond to customer inquiries, update CRM records, and advance pipeline stages without a salesperson in the loop. The agents were not helping the salespeople — they were doing the salesperson's work.

Salesforce responded to this transformation with a significant pricing change: Agentforce is priced at $2 per conversation — a per-outcome model applied to the agent's work, not a per-seat model applied to the human's access. This was not a minor commercial adjustment. It was an acknowledgment that the product category had changed: Salesforce was no longer selling CRM software. It was selling AI-delivered sales operations. The pricing change reflected the category change.

Harvey AI, founded in 2022 specifically for the legal sector, provides a different example: a company that was designed for the convergence rather than navigating it after the fact. Harvey's core product is not a legal research tool that lawyers use to research faster. It is an AI that does legal work — contract review, due diligence, regulatory research, brief drafting — on behalf of law firms and corporate legal departments. Harvey has never primarily sold access to a tool. From day one, the commercial conversation has been about the work the AI does.

Harvey's pricing model reflects this: a combination of subscription access for the platform plus consumption-based charges for specific AI work products (contract reviews, research summaries, due diligence analyses). The subscription covers access and governance; the consumption charges capture the value of the work. This two-part structure is well-suited to the convergence category because it preserves the budget predictability that enterprise legal departments need while capturing the work-based value that makes Harvey commercially defensible.

Cognition's Devin, the AI software engineer, represents the most complete expression of the convergence category. Devin does not assist software engineers — it writes, tests, and deploys code autonomously. It is a service delivered by an AI that happens to use software engineering methodology. Cognition's pricing — a subscription that includes a certain capacity of engineering work, with additional capacity available on a consumption basis — treats Devin as what it actually is: an AI service provider that happens to produce software as its output.

What these three examples share is that each company recognized the category shift and redesigned its commercial model to reflect it. The pricing is not perfect in any case — Service as Software pricing is genuinely hard to get right — but each company is at least trying to charge for the work rather than for access to the tool. That alignment between value delivered and value charged is the foundation of a sustainable commercial model in this category.

EXAMPLE: SALESFORCE / AGENTFORCE · Enterprise CRM → AI Sales Agent	
What they built	Agentforce: autonomous AI agents that qualify leads, respond to inquiries, advance pipeline stages — without a salesperson in the loop.
How the AI works	The agents use Salesforce's CRM data to identify high-value leads, craft personalized outreach, log interactions, and move opportunities through the pipeline. Human salespeople review exceptions and focus on relationship-intensive deals.
Commercial model	$2 per conversation — per-outcome pricing applied to agent work, not per-seat pricing for human access. Explicit acknowledgment that the category had changed.
Why it works	Salesforce was selling AI-delivered sales operations, not CRM software. The $2/conversation price captures the value of each commercial interaction the AI completes — not the value of having the capability available.

EXAMPLE: HARVEY AI · Legal Tech → AI Legal Service	
What they built	Harvey does legal work: contract review, due diligence analysis, regulatory research, brief drafting. Not a tool for lawyers — a service delivered to lawyers.
How the AI works	Harvey processes legal documents through fine-tuned models trained on legal precedent, identifies issues by category and severity, suggests redline language,

	estimates risk levels, and delivers structured review documents in the format legal workflows require.
Commercial model	Subscription access for platform plus consumption-based charges for specific AI work products (contract reviews, research summaries, due diligence analyses). Two-part structure: governance subscription + work product billing.
Why it works	Harvey was designed for the convergence from day one. The commercial model reflects the category: charge for the work, not for access to the capability.

EXAMPLE: COGNITION / DEVIN · Developer Tooling → AI Software Engineer	
What they built	Devin writes, tests, debugs, and deploys code autonomously. It is not a coding assistant — it is an AI software engineer that executes engineering tasks end-to-end.
How the AI works	Devin receives a task specification, breaks it into subtasks, writes code, runs tests, debugs failures, iterates to a working solution, and submits a pull request. Human engineers review the PR and set the task scope.
Commercial model	Subscription including defined engineering work capacity plus consumption charges for additional capacity. Treats Devin as an AI service provider, not a developer tool.
Why it works	The pricing model matches the product's nature: Devin is an AI service that produces software engineering work. Capacity-based subscription captures the retainer-like structure of an engineering service relationship.

Chapter One — The Essentials

› The convergence moment arrives when customers experience the AI as doing the work, not helping them do it.

› Salesforce, Harvey, and Cognition all recognized the category shift and redesigned their commercial models to reflect it.

› The $2/conversation Agentforce price is not a minor commercial adjustment — it is an acknowledgment that the product category changed.

› Companies designed for the convergence (Harvey) have a structural advantage over companies navigating to it (Salesforce).

› The question for every AI software company: has your product crossed the convergence threshold, and does your pricing model know it?

CHAPTER TWO

The Service-Software Spectrum: Where Does Your Product Sit?

From pure tool (Microsoft Word) to pure service (McKinsey) and everything between. The diagnostic.

The spectrum from pure tool to pure service is a useful diagnostic for any software company navigating the convergence. The key question is not "do we use AI?" — almost every software company uses AI now. The key question is "where does our AI sit on the spectrum, and does our pricing model reflect that position?"

At the pure tool end of the spectrum sit products where the human is fully in the loop: Microsoft Word, Google Sheets, traditional CRM systems. The human performs the task; the software is the instrument. The value proposition is capability — here is a capability you can use to do your work better. Per-seat pricing is entirely appropriate here. The value is access to the capability, and the billing unit correctly reflects that value.

Slightly further along the spectrum are AI-assisted tools: products where AI recommendations, suggestions, or automation supplement human decision-making. GitHub Copilot is a canonical example — it suggests code completions and generates code snippets, but the developer decides what to accept, what to modify, and what to reject. The developer is still doing the work; the AI is making them faster. Per-seat pricing is still defensible here, though it becomes increasingly inadequate as AI contributions to the work product grow.

In the middle of the spectrum are AI-augmented workflows: products where the AI does a substantial portion of the defined workflow, with humans reviewing, approving, and directing. Most current enterprise AI deployments fall here. The contract review AI that drafts the issues list and suggests language; the customer service AI that drafts responses for human approval; the financial analysis AI that generates the first-pass model for an analyst to refine. The human is still the decision-maker, but the AI is doing

most of the work. Per-seat pricing is increasingly misaligned here — the seats are not where the value is being created.

Further along are autonomous workflow products: products where the AI executes complete, defined workflows without human involvement in individual transactions. The support ticket AI that resolves inquiries from submission to closure, the contract review AI that reviews and annotates an entire NDAs batch overnight, the invoice processing AI that extracts data, validates against purchase orders, and posts entries to the general ledger. Humans review exceptions and monitor performance, but the workflow runs autonomously. Per-seat pricing is fundamentally misaligned here. The value is in the workflow completions, and the billing unit should reflect that.

At the pure service end of the spectrum are AI service providers: companies whose entire value proposition is delivering a professional-quality work product at scale. Harvey doing legal work. Abridge transcribing and summarizing clinical notes. Ambience Healthcare generating complete visit documentation from conversation. These companies are not selling software tools. They are selling services — delivered by AI at software economics. Their billing must reflect the service, not the software.

The diagnostic question for any software company: where does your product sit on this spectrum, and does your pricing model reflect that position? If your product is in the autonomous workflow or pure service territory, and your pricing is still per-seat or per-token, you have a commercial model misalignment. The size of that misalignment determines the size of your open-claw gap.

The Service-Software Spectrum — Five Positions				
Position	**How the AI works**	**Value proposition**	**Appropriate pricing**	**Example companies**
1. Pure tool	Human operates software; no AI or AI as grammar check	Access to capability — the human does the work	Per seat, per user	Microsoft Word, Google Sheets, legacy CRM
2. AI-assisted tool	AI supplements human decision-	Access to enhanced capability — AI	Per seat (still appropriate);	GitHub Copilot, Grammarly,

	making with suggestions	helps the human do the work faster	consumption add-on emerging	Salesforce Einstein suggestions
3. AI-augmented workflow	AI does substantial portion of defined workflow; human reviews and approves outputs	Human-AI collaboration — AI does most of the work, human decides	Hybrid subscription + per-task; per-seat increasingly inadequate	Notion AI, Jasper, most current enterprise AI deployments
4. Autonomous workflow	AI executes complete defined workflows; human monitors and handles exceptions	Workflow execution at scale — human is not involved in individual transactions	Per task, per workflow, per outcome; subscription is just a minimum commitment	Intercom AI, Harvey AI (for routine agreements), most modern AI agents
5. Pure AI service	AI delivers professional-quality work product directly; human is the customer, not the operator	Professional service at software economics — AI does the work of a professional	Per outcome, per work product, gain-share; no seat logic applies	Abridge, Ambience Healthcare, Harvey AI (for complex work)

DIAGNOSTIC QUESTION

Where is your product — and does your pricing reflect it?

For each major product you sell: (1) Describe a typical deployment after 6 months of operation. Who is doing the work — the human using the tool, or the AI? (2) What does the customer measure to evaluate whether they got value — user adoption metrics, or outcome metrics? (3) If the customer doubled their AI deployment depth without hiring additional staff, would your revenue increase? If the answer to question 1 is 'the AI', question 2 is 'outcomes', and question 3 is 'no', you are at position 4 or 5 on the spectrum with position 1 or 2 pricing. That gap is your open-claw problem.

Chapter Two — The Essentials

› The spectrum has five positions: pure tool → AI-assisted → AI-augmented → autonomous workflow → pure AI service.

› Most current enterprise AI deployments are at position 3 or 4; most are priced at position 1 or 2.

› The diagnostic reveals the gap: who does the work, what measures value, and does revenue grow with deployment depth.

› The convergence does not happen at a single moment — the spectrum is continuous, and companies drift along it as AI capability deepens.

› Designing for a specific spectrum position is a commercial choice as much as a product choice.

CHAPTER THREE

Why Existing Pricing Models Break at the Convergence Point

Seat pricing assumes access. Service pricing assumes effort. Neither fits outcome-at-scale.

Understanding precisely why existing pricing models break at the convergence point requires examining the hidden assumptions embedded in each model — assumptions that hold in the category for which the model was designed and fail in the Service as Software category.

Per-seat pricing assumes that value is proportional to access. A company with 100 salespeople gets roughly 100 times the value from CRM software as a company with one salesperson, so pricing by the seat is a reasonable proxy for value. But when the AI is doing the sales work, the number of salespeople is not what determines value — the number of qualified leads advanced, the number of opportunities created, and the number of deals closed determines value. A company that uses Agentforce to generate 1,000 qualified conversations with a ten-person sales team gets more value than a company with 100 salespeople who are not using AI at all. Per-seat pricing is measuring the wrong thing.

Per-token pricing assumes that value is proportional to consumption. For an AI tool that developers are using to write code faster, this is a reasonable proxy — the developer consuming more tokens is getting more suggestions and presumably writing more code. But for an AI agent that is reviewing contracts, the tokens consumed to review a 50-page

agreement are not the value created. The value is the reviewed agreement with issues identified and language suggested. The token count is an input cost, not an output value. Per-token pricing captures the cost, not the value.

Traditional service pricing assumes that value is proportional to effort. Consulting firms, law firms, and accounting firms traditionally charge by the hour because the effort required to deliver professional services is a reasonable proxy for the value created — more complex matters require more hours, and more hours cost more. But AI services do not have that relationship between effort and value. An AI that reviews 500 contracts in the time it took a human to review five contracts has not created 100 times the value of the human — it has created comparable value for each contract review while dramatically changing the economics. The 'effort' input is near zero; the value output is unchanged. Effort-based pricing cannot capture this.

The Service as Software category requires pricing models that measure value directly — not through proxies of access, consumption, or effort. The five models developed in Part Two of this book are designed specifically for this requirement.

Why Three Standard Models Fail at Convergence

Pricing model	Hidden assumption	Holds when?	Breaks when?	Cost of the break
Per-seat / subscription	Value proportional to number of users who have access	Human workforce is the productive unit — more seats = more work done	AI does the work without proportional headcount growth — customer gets more value but pays the same	Expansion revenue disconnected from value growth; open-claw gap widens every quarter
Per-token / consumption	Value proportional to AI resource consumption	Token consumption is a reliable proxy for value delivered	AI becomes more efficient and delivers the same value with fewer tokens; or AI delivers dramatically more value per token	Vendor's revenue falls as AI improves; or vendor underprices high-value use cases because

				the token count is low
Time-based / effort	Value proportional to effort (hours, FTE equivalent)	Professional labor cost is the dominant variable cost	AI has near-zero marginal cost per unit of professional work delivered	Pricing based on effort imputed to the AI's work misrepresents both the cost structure and the value delivered

> ***"Per-seat pricing is measuring the wrong thing. Per-token pricing is measuring the wrong thing. Time-based pricing is measuring the wrong thing. Service as Software requires pricing that measures the right thing: the work accomplished."***

Chapter Three — The Essentials

› Per-seat pricing fails because it measures access, not outcomes — and AI creates value through outcomes, not through headcount.

› Per-token pricing fails because it measures consumption, not value — and AI can deliver more value with fewer tokens as it improves.

› Effort-based pricing fails because it measures cost, not value — and AI's marginal cost approaches zero while the value it delivers does not.

› The right pricing model measures the work accomplished — the outcome, the task, the deliverable.

› Understanding precisely why each standard model fails is the prerequisite for designing a model that does not.

PART TWO

Five Pricing Models for Service as Software

Each model: definition · economics · worked example · when it works · when it fails.

CHAPTER FOUR

Model 1 — Outcome-Per-Unit Pricing

Price per resolved ticket, reviewed contract, closed deal, diagnosed case. The cleanest model — and the most demanding.

Outcome-per-unit pricing is the cleanest expression of the Service as Software commercial philosophy. You define a unit of work. The AI delivers it. The customer pays for it. Nothing about access, consumption, or effort enters the pricing equation. The commercial relationship is purely about results.

The model sounds simple. The implementation is demanding. The simplicity is in the commercial logic; the difficulty is in the precision of the unit definition.

The canonical example is the support ticket. Intercom, Zendesk, and several specialist AI customer service companies have moved toward per-resolution pricing: the customer pays a fixed amount for each support ticket that the AI resolves without human escalation. The unit is a "resolved ticket" — specifically defined as a ticket that is closed with no human agent involvement and no re-open within a defined window (typically 24–72 hours).

This definition seems obvious, but its precision took years of commercial iteration to reach. Early outcome pricing in customer service AI used vaguer definitions: "successfully handled conversation," "customer issue addressed." These definitions created billing disputes at scale — vendors believed they had handled conversations that customers believed were unresolved. The specificity matters enormously: "ticket closed, no human escalation, not re-opened within 24 hours" is a definition that both parties can evaluate unambiguously from the same data.

Consider what Harvey AI does with per-review pricing for legal work. When Harvey reviews a contract, the deliverable is a structured review document: identified issues organized by category and severity, suggested redline language for each issue, risk assessment, and a summary of recommended actions. The pricing per review is not for the tokens consumed to produce this document or for the time spent — both of which are irrelevant. The pricing is for the review document itself, with its defined structure, its identified issues, and its actionable recommendations. Each review is a unit of legal service delivered at software scale.

The economics of outcome-per-unit pricing are transformational for the vendor when properly implemented. Consider the cost structure: if Harvey's AI reviews a 50-page contract for $200, and the marginal cost of that review (model inference, infrastructure) is $8, the gross margin is 96%. If a human paralegal reviews the same contract at a billing rate of $150 per hour and takes 4 hours, the law firm charges $600. Harvey charges $200. The customer gets comparable work product at one-third the price. Harvey gets 96% gross margin. The human paralegal's labor cost is not a meaningful factor in Harvey's economics.

This economics profile — dramatically lower price for the customer, dramatically higher margin for the vendor — is the signature of Service as Software done right. It is possible because the AI's variable cost is a tiny fraction of the human's labor cost, while the output quality is comparable. The pricing model must reflect this: charging by the review captures the full value without revealing the extraordinary margin profile to the customer in a way that invites renegotiation.

The unit definition checklist for outcome-per-unit pricing has six criteria that the definition must satisfy. It must be binary: either the outcome occurred or it did not. It must be measurable: the determination of whether it occurred is based on observable data, not subjective assessment. It must be attributable: the AI's delivery of the outcome can be distinguished from the outcome occurring without the AI. It must be economically meaningful: the outcome has a quantifiable value to the customer that anchors the per-unit price. It must be completable: the AI can reliably bring the outcome

to its defined conclusion, not just initiate it. And it must be contestable only in defined ways: the definition specifies what disputes are possible (about whether the outcome occurred) and forecloses disputes that are not defined (about whether the outcome was good enough).

Intercom's approach to the resolution definition illustrates the completable criterion in practice. Intercom defines a "resolution" not just as the conversation ending but as ending with the customer's expressed satisfaction or with a follow-up action completed. Early definitions used conversation end as the proxy, which created gaming incentives — agents trained to close conversations quickly without resolution. The completable criterion forces the definition toward the actual outcome, not the process that supposedly leads to it.

Examples by sector: In healthcare, Abridge prices per clinical note completed — not per transcription, not per token, but per structured note in the required format submitted to the EHR within the defined turnaround window. In finance, Brex's AI expense categorization prices per transaction correctly categorized — a binary outcome (correct or not, as judged by a sample audit against the defined categorization rules). In recruiting, AI screening tools price per qualified candidate delivered — not per application reviewed, but per candidate who meets the defined qualification criteria and has agreed to be contacted.

MODEL 1: OUTCOME-PER-UNIT *Pay for the work. Not for the access. Not for the consumption. For the verified result.*	
Definition	A fixed price for each verified unit of work delivered — each resolved ticket, reviewed contract, processed invoice, completed analysis. The billing event is triggered by verified outcome delivery.
Billing trigger	Verification event: ticket closed without escalation + not reopened within 24h; contract review delivered + no dispute filed within 48h; invoice processed + GL entry confirmed; analysis delivered + accepted by requestor.
Economics	Gross margin = price per unit minus model inference cost minus QA cost minus SLA credit reserve. Example: $300 per contract review; $8 inference; $10 QA; $6 credit reserve = $276 gross margin = 92%.

Expansion mechanic	Revenue scales automatically with AI deployment depth. As the customer deploys the AI to more workflows, more unit completions are billed. No seat negotiation required.
When it works	Clear, binary outcome definition · Verifiable from customer system data · AI reliability sufficient to accept outcome risk · Customer comfortable with variable billing
When it fails	Ambiguous outcome definition → billing disputes · Outcomes not independently verifiable → credibility risk · AI performance below threshold → excessive credits · Customer needs budget predictability → hybrid model instead
Unit definition checklist	Binary (occurred or not) · Measurable (objective data) · Attributable (AI caused it) · Economically meaningful (quantifiable customer value) · Completable (AI brings to defined conclusion) · Contestable only in defined ways

EXAMPLE: INTERCOM · Customer Service AI	
What they built	AI-first customer service platform where the AI resolves customer inquiries end-to-end — including complex billing disputes, technical troubleshooting, and account management issues.
How the AI works	The AI reads the customer's inquiry, accesses the customer's account data, identifies the issue category, executes the resolution workflow (refund, configuration change, escalation routing), and confirms resolution with the customer. Human agents handle only the issues the AI escalates.
Commercial model	$0.99 per 'resolution' — defined as a conversation where the customer's issue was resolved by the AI without human involvement, confirmed by the conversation being closed without reopening within 24 hours. Not per conversation, not per message, not per token. Per resolution.
Why it works	The per-resolution pricing aligns vendor economics with customer value perfectly. Intercom earns revenue when the AI works. Customers pay for results, not for the AI having an opportunity to work. The pricing model is also a quality commitment: Intercom is essentially guaranteeing that what they charge for is a genuine resolution, not just a closed conversation.

EXAMPLE: HARVEY AI · Legal Tech	
What they built	AI-powered legal work platform for contract review, due diligence, regulatory analysis, and brief preparation at law firms and corporate legal departments.
How the AI works	Harvey processes legal documents through models fine-tuned on legal precedents and firm-specific playbooks. It identifies issues by category and severity, suggests contractual language alternatives, estimates risk levels, and delivers a structured review in the format the legal team's workflow requires.

Commercial model	Per-review pricing for specific document types: $150–$500 per NDA review, $400–$2,000 per complex commercial agreement review, depending on document length and review depth. Pricing is per completed review document, not per page or per token.
Why it works	The per-review pricing allows Harvey to capture the value of legal work delivered (a human paralegal would charge $300–$800 for comparable work) while maintaining extraordinary economics (AI inference + QA cost < $30 per review). The pricing model also creates a natural quality test: customers who receive reviews that miss material issues will dispute and not pay — providing automatic quality feedback.

Chapter Four — The Essentials

› Outcome-per-unit is the cleanest alignment between value delivered and price charged — and the model with the highest ceiling.

› The unit definition must satisfy six criteria: binary, measurable, attributable, economically meaningful, completable, and contestable only in defined ways.

› Intercom's $0.99 per resolution is a textbook implementation: specific definition, customer-system verification, no ambiguity.

› Harvey's per-review pricing captures legal service value at software economics — 90%+ gross margins on $300–$2,000 per-unit work.

› The most common failure mode: ambiguous outcome definition. Invest in definitional precision before commercial launch.

CHAPTER FIVE

Model 2 — SLA-Committed Capacity Pricing

Buy X agent-hours per month guaranteed. The service retainer reimagined for AI.

SLA-committed capacity pricing is the service retainer reimagined for AI. Instead of buying access to a tool (the SaaS subscription model) or paying for specific outcomes (the outcome-per-unit model), the customer buys a defined capacity of AI service

delivery — and the vendor commits to delivering that capacity to a defined performance standard.

The model is structured as a retainer: the customer commits to a monthly or annual fee in exchange for the vendor's commitment to provide a defined volume of AI service work, available on demand, at a defined quality level. The retainer is not a subscription — it is a service level agreement with financial consequences for non-performance.

The best analogy is the law firm retainer. A corporate legal department that retains a law firm for $50,000 per month is not buying access to the lawyers — it is buying guaranteed availability of legal expertise at a defined service level (response time, volume of work, quality standard). If the law firm fails to deliver the committed service level, the retainer arrangement is in breach. The Service as Software equivalent makes the same commitment, with the AI doing the work instead of the lawyers.

Consider how this would work for a contract review AI. The customer commits to $30,000 per month. In exchange, the vendor commits to: reviewing up to 200 standard contracts per month (NDAs and standard commercial agreements up to 20 pages), with a 4-hour turnaround from submission to reviewed document, with issue identification at a recall rate above 95% (validated by quarterly sample audit against expert human review), and with availability of 99.5% during business hours. If the vendor fails to meet the 200-review capacity or the quality standard in any month, a credit formula applies — typically a percentage of the monthly fee proportional to the shortfall.

The key distinction from the outcome-per-unit model is the commitment structure. In outcome-per-unit, the customer pays only for what the AI delivers; if volume is low in a month, the bill is low. In SLA-committed capacity, the customer pays for committed availability regardless of whether they use the full capacity; if they submit only 100 contracts in a month where 200 were committed, they have still paid for the capacity.

This distinction matters significantly to the enterprise buyer. CFOs and procurement teams can budget for a retainer far more easily than they can budget for a variable per-outcome payment. The retainer maps directly to their existing mental model of

professional services procurement. This is one reason why the capacity model often unlocks larger commitments than the per-outcome model — the customer is more comfortable making a larger fixed commitment than a variable commitment of unknown total cost.

Ironically, the capacity model is often better for the vendor's economics as well. Under pure per-outcome pricing, the vendor takes full capacity risk: if the customer submits 50 contracts instead of 200 in a month, the vendor earns one-quarter of the expected revenue while the infrastructure cost remains roughly constant. Under the capacity model, the vendor earns the full retainer regardless of actual volume, transferring the capacity utilization risk to the customer.

The operational requirement for the capacity model is a reliable capacity management system: the vendor must be able to guarantee the committed capacity and must have the infrastructure to sustain it even when demand spikes above the committed level. This is not trivial for AI services with variable inference costs — the vendor must size their infrastructure for the committed capacity plus a buffer, not just for average actual volume.

A sophisticated version of the capacity model used by several enterprise AI vendors adds a burst provision: the committed capacity is available as a base, and additional capacity above the commitment is available at a premium rate with a longer turnaround SLA. This structure gives customers certainty on their committed capacity while creating a natural expansion mechanism — customers who regularly need burst capacity are candidates for a retainer increase.

Real-world examples: Abridge Healthcare offers SLA-committed capacity to health systems — a defined number of clinical notes per month, with guaranteed turnaround time and minimum accuracy standards, priced as an annual commitment. Health systems can budget the commitment, project the labor savings, and present a business case to the CFO with a clear ROI profile. The per-note alternative pricing would have made budgeting harder and the CFO conversation less clean.

Glean, the enterprise AI search platform, has moved toward an SLA-committed capacity model for its AI agents that proactively surface information and draft documents. Enterprise customers commit to an annual capacity for AI-generated work products, with the commitment structured around the types of work (research summaries, meeting preparation, document drafts) rather than just a token volume. The specificity of the capacity definition makes the commitment's value more tangible to the buyer.

MODEL 2: SLA-COMMITTED CAPACITY

Guaranteed availability. Defined performance. The retainer structure that enterprise finance can approve.

Definition	A committed monthly or annual fee that buys guaranteed AI service capacity at a defined quality level. The fee is owed regardless of actual utilization within the committed capacity.
Commitment structure	Volume commitment: X units of work per period. Quality commitment: minimum recall/precision/accuracy at defined SLA. Availability commitment: percentage uptime during business hours. Turnaround commitment: maximum time from submission to delivery.
Economics	Revenue is the retainer fee, predictable regardless of actual volume (within the committed capacity). COGS is capacity infrastructure cost plus QA cost, sized for committed capacity plus buffer. Underutilized capacity improves vendor margins; overutilized capacity triggers burst pricing.
Customer value	Budget predictability — the CFO can include the retainer as a fixed line item. Guaranteed availability — the AI is always available for committed work without queuing. Relationship structure — retainer implies ongoing commitment from both sides.
When it works	Enterprise buyers with predictable AI work volumes · CFOs who need fixed budget lines · Use cases where availability guarantee matters · Customers who value the accountability of a service relationship
When it fails	Highly variable work volumes (customer may pay for unused capacity) · Work volumes far exceeding the commitment (forces expensive burst pricing) · Customers resistant to commitments before AI performance is proven
Burst provision	Above committed capacity: available at premium rate (typically 30–50% above the effective per-unit rate at committed volume) with a longer SLA turnaround. Creates natural expansion conversation.

EXAMPLE: ABRIDGE · Healthcare AI — Clinical Documentation

What they built	AI clinical documentation platform that automatically generates clinical notes, after-visit summaries, and care plans from physician-patient conversations.
How the AI works	Abridge captures the physician-patient conversation in real time, processes it through models fine-tuned on clinical documentation requirements, generates a complete SOAP note in the required format, and submits it to the EHR system — all within minutes of the visit ending. The physician reviews, edits minimally if needed, and signs.
Commercial model	Annual capacity commitments to health systems: defined number of clinical notes per month (matched to the health system's typical patient volume), with guaranteed turnaround time (typically note available in EHR within 5 minutes of visit end), minimum accuracy standard (verified by quarterly clinical accuracy audits), and 99.5% uptime during clinical hours. Health systems budget the annual commitment.
Why it works	Health systems need predictability. They cannot budget for a per-note variable cost model any more than they can budget for unpredictable physician compensation. The capacity retainer maps to how health systems think about service contracts. The annual commitment also gives Abridge revenue certainty to invest in the infrastructure required to serve large health systems.

EXAMPLE: GLEAN · Enterprise AI Knowledge Work	
What they built	Enterprise AI platform that proactively surfaces relevant information, generates research summaries, prepares meeting briefs, and drafts documents based on each employee's context and role.
How the AI works	Glean's AI agents continuously index the enterprise's knowledge base, monitor each employee's work context (calendar, emails, documents in progress), and proactively surface relevant information. When asked, they generate complete research summaries, meeting preparation documents, or first drafts — not search results, but actual work products.
Commercial model	Annual capacity commitment by work product type: X research summaries per month, Y meeting preparation documents per month, Z document first drafts per month. Each work product type has defined quality standards and turnaround SLAs. Committed volume below typical usage means most customers stay within the commitment; occasional excess is handled by burst pricing.
Why it works	The work product commitment structure is more commercially precise than token-based pricing for enterprise buyers who need to evaluate ROI. A CFO can evaluate 'we pay $20,000 per month for 500 research summaries and 200 meeting briefs' far more easily than they can evaluate '$20,000 per month for X million tokens'.

Chapter Five — The Essentials

› SLA-committed capacity is the service retainer reimagined for AI — guaranteed availability, committed volume, defined quality standard.

› Enterprise finance can approve retainers easily: they map to existing professional services procurement patterns.

› The capacity model transfers utilization risk to the customer — which is commercially acceptable when the customer values availability guarantees.

› Abridge's health system contracts demonstrate that complex institutional buyers will commit to capacity-based pricing when the value is clear and the procurement pattern is familiar.

› The burst provision above committed capacity is the natural expansion mechanism — heavy users are candidates for retainer increases at renewal.

CHAPTER SIX

Model 3 — Tiered Outcome Quality Pricing

Standard vs Premium vs Expert tiers of AI service quality. Differentiation without differentiation.

Tiered outcome quality pricing addresses a specific commercial challenge in Service as Software: how do you differentiate your offering when you are competing in a market where all the AI systems are using the same foundation models?

The challenge is real. When OpenAI, Anthropic, and Google all offer API access to models of comparable capability, and when most AI application companies are building on top of these models, the raw model capability is becoming commoditized. A legal AI built on Claude is not fundamentally differently capable from a legal AI built on GPT-4o. Both can review contracts. Both can identify issues. Both can suggest language.

The differentiation must therefore come from something other than raw model capability: the training data, the domain expertise embedded in the prompts and

evaluation frameworks, the user interface and workflow integration, the data security and compliance posture, and — critically — the quality guarantee.

Tiered outcome quality pricing creates differentiation by packaging these non-model elements into distinct tiers that offer different quality guarantees. The Standard tier uses a capable but cost-efficient approach, delivers results within a standard turnaround time, and offers a basic quality guarantee (issue identification above a defined recall threshold). The Professional tier uses a more thorough analysis methodology, delivers faster, and offers a higher quality guarantee with a more stringent recall standard. The Expert tier uses the most thorough methodology available, with human-in-the-loop review for complex items, and offers the highest quality guarantee.

This tiering logic is borrowed directly from professional services, where the senior partner charges more than the associate not because they work harder but because their judgment is better and their output is more reliable. The AI service equivalent: the Expert tier is more expensive not because it consumes more tokens but because it delivers more reliable output at a higher quality standard with greater accountability.

Consider how a contract review AI might implement three-tier quality pricing. The Standard tier reviews contracts using the base AI model with standard prompts, identifies issues above a 0.5 severity threshold, and delivers within 24 hours. Price: $150 per contract. The Professional tier reviews contracts using an enhanced methodology with domain-specific fine-tuning, identifies issues above a 0.3 severity threshold (catching more subtle issues), and delivers within 6 hours. Price: $450 per contract. The Expert tier reviews contracts using the Professional methodology plus human attorney review of the AI's issue identification, provides counsel-level commentary on risk allocation, and delivers within 24 hours with attorney certification. Price: $1,200 per contract.

This pricing structure does several things simultaneously. It preserves the market for the existing standard-tier offering while creating significant upside from customers whose use cases require higher quality. It creates a natural upgrade path as customers' trust in the AI's output grows — they may start on Standard to build confidence, then

upgrade to Professional as they deploy for higher-stakes agreements. It also creates a competitive moat: the Expert tier with human-in-the-loop review is difficult to replicate without the operational infrastructure to manage the human review component at scale.

The economics of the tiered model reward the vendor for the quality differentiation investments that competitors without these investments cannot match. A commoditized AI service that can only compete on Standard-tier pricing is in a race to the bottom. A tiered model with a credible Expert tier above the commodity pricing is competing on quality, not just price.

Navan, the corporate travel and expense management platform, uses a version of tiered quality pricing for its AI expense categorization. The Standard tier handles routine expense categorization automatically. The Professional tier applies enhanced policy compliance checking and flags edge cases. The Premium tier adds real-time human review of flagged items and CFO-grade audit reports. Enterprise customers with complex expense policies and regulatory requirements pay meaningfully more for the Professional and Premium tiers — because the quality difference at their compliance requirements is commercially significant.

MODEL 3: TIERED OUTCOME QUALITY *Sell quality, not just capability. The tier is a quality guarantee, not just a feature bundle.*	
Definition	Multiple pricing tiers for the same fundamental capability, differentiated by quality guarantee (accuracy standard), thoroughness (analysis depth), speed (turnaround SLA), and accountability (human-in-loop review at higher tiers).
Why it matters	When underlying model capability is commoditized, quality differentiation is the only sustainable competitive moat. The tier price reflects the vendor's confidence in the quality they commit to at each level.
Tier 1 — Standard	Base AI model · Standard thoroughness · Standard turnaround (24h) · Minimum quality guarantee (e.g., 85% recall) · Software-level pricing
Tier 2 — Professional	Enhanced methodology + domain fine-tuning · Higher thoroughness · Faster turnaround (4–8h) · Higher quality guarantee (e.g., 95% recall) · Premium price (2–4× Standard)

Tier 3 — Expert	Professional methodology + human expert review · Maximum thoroughness · Certified output · Highest quality guarantee (e.g., 99% recall) · Professional services pricing (5–8× Standard)
Expansion mechanic	Customers start on Standard for low-risk use cases; upgrade to Professional or Expert as they deploy AI for higher-stakes work. Tier upgrades are natural commercial conversations grounded in quality requirements.
When it works	Markets where quality variance is meaningful to buyers · Use cases with clearly differentiable quality levels · Sectors where professional accountability has commercial value (legal, medical, financial)

EXAMPLE: CONTRACT REVIEW AI — HYPOTHETICAL DESIGN · Legal Technology	
What they built	A contract review AI offering three quality tiers for different contract types and risk tolerances.
How the AI works	Standard: Base AI reviews using standard legal issue detection, 24-hour delivery, 85% recall on issues of severity ≥ 3 on a 5-point scale. Professional: Enhanced review using domain-specific fine-tuning on jurisdiction and contract type, 6-hour delivery, 95% recall on severity ≥ 3 issues and 80% recall on severity ≥ 2. Expert: Professional review plus human attorney verification of all severity ≥ 3 issues, attorney certification, 24-hour delivery, 99% recall guarantee on severity ≥ 3.
Commercial model	Standard: $150 per contract (NDAs, standard MSAs). Professional: $450 per contract (complex commercial agreements, employment contracts). Expert: $1,200 per contract (M&A ancillary agreements, regulated sector contracts, high-value transactions).
Why it works	The tier structure addresses the 'we need professional quality but at software price' tension that enterprise legal departments face. Standard handles the commodity volume efficiently. Professional handles the routine complex work at a reasonable premium. Expert provides accountability for the highest-stakes work that would otherwise require outside counsel.

EXAMPLE: NAVAN · Corporate Travel and Expense AI	
What they built	AI-powered expense and travel management platform with tiered AI-assisted expense intelligence.
How the AI works	Standard: Automatic expense categorization using AI rules engine, basic policy compliance checking. Professional: Enhanced categorization with exception flagging and manager notification for policy violations, integration with ERP cost centers. Premium: Real-time policy compliance with human review of flagged items, CFO-grade audit reports, custom expense policy implementation.
Commercial model	Standard included in base subscription. Professional: $5/employee/month premium. Premium: Custom pricing based on employee count and compliance

	requirements. Premium tier customers include publicly-listed companies with SOX compliance requirements and PE-backed portfolio companies with financial reporting obligations.
Why it works	The tiered structure captures meaningful price differentiation based on compliance requirements. Companies that need audit-grade expense documentation pay a premium for the Expert tier's human review and certification — a premium that is small compared to the cost of manual compliance processes it replaces.

Chapter Six — The Essentials

› Tiered outcome quality is the anti-commoditization strategy for Service as Software — compete on accountability, not just capability.

› The quality guarantee differentiates the tiers more than features do: customers pay more for a higher recall guarantee than for additional features.

› The Expert tier with human-in-loop review creates a moat that pure-AI competitors cannot easily replicate.

› Legal, healthcare, and financial services are the primary sectors where tiered quality pricing commands significant premiums.

› The tier structure creates a natural upgrade path as customers move AI deployment to higher-stakes use cases.

CHAPTER SEVEN

Model 4 — Gain-Share and Risk-Sharing

Vendor and customer share the upside. The consulting engagement model applied to AI agents.

Gain-share and risk-sharing is the pricing model that most completely aligns vendor economics with customer value creation. The vendor charges a base fee for the AI service, plus a percentage of the measured value delivered above a defined baseline. If the AI delivers exceptional results, the vendor earns exceptional revenue. If the AI underperforms, the vendor earns less.

This model is the commercial equivalent of a consulting engagement with outcome-based success fees — a structure that has existed in professional services for decades but that is rarely attempted in software because software vendors historically could not measure their impact on customer outcomes with enough precision to make outcome-based pricing credible.

AI services change this calculation. Because AI services are deeply integrated with customer workflows and data systems, they can generate the measurement data required for gain-share pricing. An AI that processes invoices can measure the error rate before and after AI deployment. An AI that generates sales content can measure conversion rates. An AI that optimizes supply chain routing can measure cost-per-mile before and after deployment. The measurement capability exists. The question is whether the vendor has the commercial confidence to stake their revenue on it.

The companies that have implemented gain-share models for AI services share several characteristics. They have deployed AI in use cases where the outcome is quantifiable in dollar terms: cost reduction, revenue increase, error rate reduction, time savings at a known labor rate. They have established baseline measurements before AI deployment so the improvement is attributable. They have agreed with customers on the measurement methodology before the pricing model is signed — disputes about gain-share calculations are commercially catastrophic if the measurement basis has not been established in advance.

Automation Anywhere and UiPath have offered gain-share variants for their RPA and AI automation deployments. The structure typically involves: a baseline measurement period (60–90 days before deployment, measuring the process metrics that will be used to calculate gain), a deployment and measurement period (12 months of AI-assisted operation with continuous measurement), a gain calculation (the dollar value of improvement in the measured metrics attributable to the AI deployment), and a gain-share payment (a percentage of the calculated gain, typically 15–30%, paid annually or quarterly). The base service fee is lower than traditional software pricing — the vendor

accepts lower certain revenue in exchange for the upside of participating in the value they create.

The risk-sharing dimension — the vendor accepting performance risk — is commercially significant beyond its economic effect. When a vendor proposes gain-share pricing, they are making an implicit statement: we are confident enough in our AI's performance to put our revenue at risk on it. This confidence is commercially persuasive in ways that feature lists and demo performances are not. The customer's procurement team, the CFO, and the technical evaluators are all more likely to trust a vendor who is willing to stake their revenue on the outcome than one who is not.

The contract structure for gain-share arrangements requires careful attention to three elements. The baseline must be mutually agreed, documented, and measured before AI deployment — an independently audited baseline is stronger than a vendor-stated one. The measurement methodology must specify exactly what is being measured, how it is measured, what data sources are used, and who adjudicates disputes. The gain attribution must address the counterfactual question — what would the outcomes have been without the AI? — typically through control group analysis, statistical modeling, or a percentage attribution agreed in advance.

Salesforce's Einstein Revenue Intelligence uses a version of gain-share pricing that shares the value of pipeline improvements attributable to AI recommendations. Customers who can demonstrate that their sales team's win rate or pipeline velocity improved after Einstein deployment receive a pricing adjustment that reduces their subscription cost — effectively a retroactive gain-share that rewards customers whose results justify attribution. This structure builds loyalty and creates a natural case study pipeline: customers whose results are good enough to claim the adjustment are customers whose success stories Salesforce can market.

The practical limitation of gain-share pricing is the measurement complexity. Not every AI service operates in an environment where the outcome is cleanly measurable and attributable. An AI writing assistant that improves the quality of customer communications creates real value, but measuring that value in dollar terms and

attributing it cleanly to the AI is genuinely difficult. Gain-share pricing is best suited to processes where the outcome is already being measured (because the business cares about it), where the measurement is objective (not requiring subjective quality assessment), and where the AI's contribution to the measured outcome is causally clear.

MODEL 4: GAIN-SHARE *Put your revenue at risk on your AI's performance. Nothing communicates confidence like shared stakes.*	
Definition	Base service fee plus a percentage of the measured economic value delivered above an agreed baseline. If the AI delivers exceptional results, the vendor earns exceptional revenue. If it underperforms, the vendor earns less.
Structure	Baseline measurement period (60–90 days pre-deployment) · Deployment + measurement period (typically 12 months) · Gain calculation (improvement in defined metrics × economic value) · Gain-share payment (vendor's percentage of calculated gain, typically 15–30%)
Credibility signal	Proposing gain-share is a statement: 'We are confident enough in our AI to put our revenue at risk on it.' This is one of the most commercially persuasive statements an AI vendor can make.
Measurement requirements	Quantifiable outcome in dollar terms · Objective measurement (not requiring subjective quality assessment) · Clean attribution (AI's contribution causally identifiable) · Agreed methodology before deployment (not litigated after)
When it works	Processes already measured by the customer · Clear causal relationship between AI deployment and outcome · Customer and vendor willing to invest in baseline measurement · High-trust, strategic relationships
When it fails	Unmeasurable outcomes · Multiple contributing factors (attribution dispute) · Short relationship (insufficient trust for risk sharing) · Customer unwilling to share outcome data required for calculation
Contract requirements	Independently audited baseline · Agreed measurement methodology and data sources · Attribution methodology (agreed percentage or statistical approach) · Dispute resolution mechanism for gain calculation disputes

EXAMPLE: AUTOMATION ANYWHERE · RPA and AI Process Automation	
What they built	Enterprise automation platform using RPA and AI to automate complex business processes — accounts payable, HR operations, supply chain, customer service.

How the AI works	Automation Anywhere's AI bots execute entire business processes end-to-end: extracting data from invoices, validating against purchase orders, posting to ERP systems, managing exceptions, and generating audit reports — all without human involvement in individual transactions.
Commercial model	Gain-share model for large deployments: baseline measurement of process metrics (error rate, processing time, cost per transaction) during a pre-deployment period. After deployment: monthly measurement of same metrics. Annual gain calculation: (improvement in metrics × unit economics) × 20–25% gain-share. Base service fee at reduced subscription rate. Typical outcome for customer: $5M in annual process efficiency savings; Automation Anywhere earns $1–1.25M in gain-share payments in addition to the base subscription.
Why it works	The gain-share model addresses the CFO's primary concern: ROI. By committing to a gain-share structure, Automation Anywhere is saying 'we will prove our value through your own financial metrics.' The measurement infrastructure also creates a continuous stream of ROI evidence that justifies contract renewal and expansion at each annual review.

EXAMPLE: SALESFORCE EINSTEIN REVENUE INTELLIGENCE · CRM AI	
What they built	AI platform that analyzes sales team activity, pipeline data, and market signals to generate AI-recommended actions and forecasts.
How the AI works	Einstein monitors every salesperson's activities, identifies pipeline risks and opportunities, recommends specific next best actions, and adjusts revenue forecasts based on AI analysis of deal health. Outcome metrics: pipeline conversion rate, forecast accuracy, win rate.
Commercial model	Einstein Revenue Intelligence includes a pricing adjustment mechanism that reduces subscription cost for customers who can demonstrate pipeline improvement attributable to AI recommendations. Not a formal gain-share, but a retroactive value-based adjustment. Customers with documented 15%+ win rate improvement receive a 10% subscription credit at renewal.
Why it works	The adjustment mechanism creates a success story pipeline: customers who qualify for the credit have results worth marketing. It also creates a retention mechanism: customers who are approaching the improvement threshold at renewal time are motivated to continue deployment to qualify for the credit.

Chapter Seven — The Essentials

› Gain-share is the most commercially persuasive pricing model — it is a confidence signal that no competitor can replicate without comparable AI performance.

› The three prerequisites: quantifiable outcome, clean attribution, agreed measurement methodology before deployment.

› Automation Anywhere's gain-share model turns the CFO conversation from 'prove your ROI' to 'here is a structure where we share in the ROI we create'.

› The baseline measurement period is non-negotiable — without an independently audited baseline, the gain calculation will be disputed.

› Gain-share is not appropriate for every AI service — identify the specific use cases where all three prerequisites are met before proposing this structure.

CHAPTER EIGHT

Model 5 — Subscription Floor Plus Outcome Overage

Predictable base plus elastic outcome billing. The hybrid that most enterprises can approve.

The subscription floor plus outcome overage model is the pragmatic bridge between the old commercial world and the new one — the structure that allows a software company to maintain the budget predictability that enterprise customers need while beginning to capture the outcome-based value that Service as Software delivers.

The model has two components. The subscription floor is a committed monthly or annual fee that gives the customer a baseline service level — a defined volume of AI work delivered at a defined quality standard. The floor provides the customer with a predictable budget line and provides the vendor with a revenue floor that covers the cost of the committed capacity. The outcome overage is a per-unit charge for AI work delivered above the subscription floor — each additional contract reviewed, each additional support ticket resolved, each additional invoice processed above the committed volume at a defined per-unit rate.

This structure is the commercial design that most enterprise customers can approve. The subscription floor maps to their existing procurement processes: it is a software subscription, it can be approved through the normal SaaS procurement workflow, it can be included in the annual IT budget cycle. The outcome overage is familiar from SaaS

consumption models: it is the equivalent of a cloud computing cost that scales with usage. Enterprise CFOs understand this structure. They can model it. They can budget for it with reasonable confidence.

The strategic value of this model for the vendor is that it creates a path from pure subscription to pure outcome pricing without the commercial disruption of a complete pricing model change. A company can introduce the hybrid model as an upgrade option from pure subscription — the customer gets a lower floor cost with upside potential if their AI usage is high, or the same floor with outcome overage replacing the old per-seat expansion model.

Critically, the hybrid model builds the operational infrastructure for eventual outcome pricing: the vendor must instrument the product to measure outcomes, must design the billing system to track and invoice overages, and must establish the operational processes to manage outcome disputes. Once this infrastructure is in place, the migration to a higher proportion of outcome pricing is straightforward.

Consider how this works in practice for a coding agent product. The subscription floor: $15,000 per month, covering up to 500 AI-completed code reviews (defined as: pull request reviewed, issues identified, suggested fixes provided, review completed within 4 hours). The outcome overage: $35 per code review above the 500-review base, same definition, same quality standard. A development organization that averages 400 reviews per month pays $15,000. One that averages 800 reviews per month pays $15,000 + (300 × $35) = $25,500.

The expansion mechanic is natural: as the customer's development velocity grows and they rely more heavily on the AI for code review, their consumption grows above the floor and the vendor captures that growth automatically. The customer's procurement team approved a $15,000 subscription; the additional charges are consumption-based and typically can be approved at a lower authority level.

The floor design is the critical commercial decision. Set it too high and light users pay for capacity they never use — they will churn or demand a lower tier. Set it too low and

the vendor underprices the base service while heavy users generate the majority of revenue in the overage — which creates renewal risk if heavy users decide the overage cost is too high. The right floor is calibrated to the p50 consumption of the target customer segment: the median customer in the segment consumes roughly what the floor provides, so the median customer pays the floor price and the heavy users pay meaningfully more.

Anthropic's Claude.ai Pro and Teams subscriptions use a version of this model: a subscription that includes a defined usage level, with access to higher usage for users who need it. The Teams tier adds organizational features and higher usage limits. This is not quite outcome pricing — it is still usage/access pricing — but the structure is the same: a floor that covers most customers' typical needs with expansion available at additional cost.

Cursor, the AI coding environment, uses a subscription that includes a monthly credits allocation for AI-assisted coding, with additional credits purchasable when the allocation is exhausted. The floor is defined in credits rather than specific outcomes, which is a less pure form of the hybrid model — but the commercial logic is identical: predictable floor, consumption expansion.

MODEL 5: SUBSCRIPTION FLOOR + OUTCOME OVERAGE *The pragmatic bridge. Enterprise finance approves the floor. AI performance drives the overage.*	
Definition	Committed monthly/annual subscription that includes a defined baseline of AI work, plus per-unit charges for work above the baseline. The floor provides budget predictability; the overage captures expansion value.
Floor design	Set at p50 consumption of target segment: median customer pays the floor and no more. Heavy users pay meaningfully above. Light users have unused capacity but remain satisfied because they pay the predictable amount they budgeted.
Overage design	Per-unit price typically 15–25% above the effective per-unit rate implied by the floor. Overage is priced as a premium to reflect the value of elastic capacity. Must feel proportional — punitive overage pricing drives customers to negotiate higher floors.

Expansion mechanic	Natural: as AI deployment depth grows, more work is completed above the floor. Revenue growth happens automatically without seat renegotiation. The expansion conversation becomes 'would you like to increase your floor to reduce your overage charges?'
Enterprise procurement advantage	Finance can approve the floor as a software subscription (standard procurement). Overage charges are consumption-based (familiar from cloud services). Both components fit existing procurement frameworks.
Migration path	The hybrid is the stepping stone: companies can migrate from pure subscription to this model (introducing per-unit overage) and then from this model to per-outcome primary (increasing the floor or eliminating it as a separate line).
When it works	Enterprise customers requiring budget predictability · Products with varying usage intensity across the customer base · Companies migrating from subscription to consumption models · Markets where pure outcome pricing hasn't yet established buyer comfort

EXAMPLE: CURSOR · AI-Native Code Editor	
What they built	AI coding environment that combines an IDE with an AI that writes, refactors, debugs, and explains code — acting as a collaborative coding partner.
How the AI works	Cursor's AI understands the entire codebase, writes code in context (not just completions), explains existing code, identifies bugs, and refactors code blocks. It processes a running context of the open project files to generate contextually appropriate outputs.
Commercial model	Subscription tiers with included fast request allowances plus ability to purchase additional requests when the allocation is exhausted. Pro plan: $20/month with 500 fast requests included. Additional fast requests available at $0.04 each. The allocation maps roughly to moderate AI-assisted coding usage; heavy users exceed it monthly.
Why it works	The subscription floor provides individual developers and teams with a predictable budget line for AI coding assistance. The overage preserves access to the AI without interruption when usage is heavy. The per-request overage price is low enough to be adopted frictionlessly but high enough to generate meaningful additional revenue from power users.

EXAMPLE: ANTHROPIC CLAUDE.AI TEAMS · Enterprise AI Assistant	
What they built	Team-level Claude deployment with collaboration features, custom instructions, and higher usage limits for enterprise teams.
How the AI works	Teams get persistent Claude projects, shared custom instructions, team-level conversation management, and higher message limits than individual

	subscriptions. The AI assists with analysis, writing, research, coding, and complex reasoning tasks across the team.
Commercial model	Teams subscription: $25/user/month with higher usage limits than Pro. Usage above the allocated limits is throttled rather than charged, but the Teams plan includes substantially higher limits that most teams do not exceed. Enterprise plan provides unlimited usage. The model is subscription floor (Teams) with effectively unlimited overage (Enterprise) for high-usage teams.
Why it works	The Teams/Enterprise pricing structure captures the full range of enterprise buyers: teams with predictable moderate usage pay the Teams floor; teams with heavy or variable usage upgrade to Enterprise for unlimited capacity. The upgrade conversation is natural because Teams customers who hit their limits experience it directly.

Chapter Eight — The Essentials

› The subscription floor plus outcome overage is the pragmatic bridge to outcome pricing — enterprise finance approves it, AI performance drives the economics.

› Floor design is critical: set at p50 customer segment consumption so median customers pay the floor and heavy users pay meaningfully more.

› The overage price should feel proportional — punitive overage drives customers to negotiate higher floors, eliminating the consumption upside.

› Cursor's $0.04 per request overage demonstrates the right pricing philosophy: accessible enough to not disrupt workflow, meaningful enough to generate revenue from power users.

› The hybrid is a migration tool as well as a pricing model — companies can evolve toward higher outcome proportions over time from this structure.

PART THREE

Contracts, Revenue Recognition, and Operations

The commercial infrastructure that makes outcome-based commitments real.

CHAPTER NINE

Contracting for Service as Software: New Clauses, New Risks

SLA design. Liability for AI errors. IP ownership of agent outputs. Force majeure for model deprecation.

Contracting for Service as Software requires clauses that do not exist in standard SaaS agreements and that most enterprise legal teams have not encountered before. The contract must specify what the AI is committing to deliver — not what the software can do, but what the service will actually produce. This shift from capability specification to commitment specification is the defining challenge of Service as Software contracting.

Five specific contract dimensions require AI-specific attention.

SLA design for AI performance is the most important and the most novel. A traditional software SLA specifies availability: the system will be accessible 99.9% of the time. A Service as Software SLA must additionally specify quality: the AI will identify at least 95% of material issues in reviewed contracts (recall rate), with fewer than 5% of identified issues being false positives (precision rate), within a 4-hour turnaround time. These quality commitments are not standard contract clauses. They require: defining the metric precisely (what is a "material issue"?), specifying the measurement methodology (how is recall measured? by whom? against what reference?), establishing the audit process (who conducts the quality audit? how frequently?), and defining the remedy (what credits apply for SLA misses?).

Consider how Harvey AI structures SLA commitments for contract review. The SLA specifies: recall rate of at least 90% for issues of severity 3 or above on Harvey's defined issue severity scale, measured monthly by random sampling of 20 reviewed contracts against an independent review by a senior attorney at a named reference law firm, with Harvey providing the independent attorney review at Harvey's cost. If the recall rate in any month falls below 90%, Harvey provides a credit equal to 10% of that month's fees.

If it falls below 80% for two consecutive months, the customer has the right to terminate for cause with a full refund of unused prepaid fees.

This SLA structure is significant for several reasons. The measurement basis (independent attorney review) removes Harvey's conflict of interest in measuring its own performance. The specific reference law firm as the measurement standard creates a consistent benchmark. The two-tier remedy (credit for modest miss, termination for sustained miss) reflects the commercial reality that occasional underperformance should be remediated financially while systematic underperformance should allow the customer to exit.

Liability for AI errors is the second novel dimension. Traditional software contracts limit the vendor's liability for the customer's use of the software — if the customer uses Word to write a misleading document, Microsoft is not liable. But when the AI is providing a service — when Harvey is delivering contract reviews that the customer's legal team relies on — the vendor's liability exposure is different in kind. A missed material issue in a reviewed contract that the customer signed in reliance on Harvey's review creates a potential professional services liability, not just a software product liability.

Most AI service companies currently limit their liability for AI errors to the fees paid for the specific service in which the error occurred, plus a cap at total annual fees. This limitation is commercially defensible for current AI capabilities — the AI is not yet reliable enough to warrant professional liability insurance at the level of human professionals. But it creates a significant gap between the liability the AI service company accepts and the liability the customer implicitly assumes when they rely on AI-generated work product.

The contracting approach that best manages this tension is to explicitly define reliance standards in the contract: the customer may use the AI's work product as a starting point for their own professional judgment, but not as a substitute for it. This clause limits the vendor's liability while establishing a clear expectation about how the AI's output should be used. It is the equivalent of a medical AI's "clinical decision support" framing — the AI supports the doctor's judgment but does not replace it.

IP ownership of AI outputs is the third novel dimension. When Harvey AI reviews a contract and generates a structured issues list with suggested redline language, who owns that issues list? The customer who provided the contract? Harvey, whose AI generated the review? The law firm attorney whose historical work trained the model? In practice, most AI service contracts assign the output to the customer — the customer pays for and owns the specific review document that the AI generates. Harvey retains ownership of its models, its prompts, and its methodology.

This standard allocation is generally correct but raises a complexity when the AI's output quality depends on the customer's data. An AI contract review system that is fine-tuned on a corporate legal department's previous contracts and issues will be more accurate for that corporate legal department's contract types than for a different organization's. The fine-tuned model incorporates the customer's data in a way that creates genuine uncertainty about data ownership and usage rights. The contract must address: what data does the vendor use for training? Does the customer's usage data improve models that are then used for other customers? Can the customer require deletion of their data, and what happens to model improvements derived from their data?

Force majeure for model deprecation is the fourth novel dimension, and the most specific to AI. Foundation models are deprecated. GPT-3.5 has been deprecated; GPT-4 was replaced by GPT-4o; future models will be deprecated in turn. When a Service as Software vendor's capability depends on a specific model version, and that model is deprecated, the service level may change. The contract must address: what happens if the underlying model changes? Does the vendor commit to maintaining service quality through model transitions? What notice must the vendor provide before changing the underlying model? What remedies does the customer have if service quality deglines after a model change?

The practical approach is to commit to output quality (the SLA standard defined above) regardless of the underlying model, with the vendor bearing the responsibility of maintaining that quality through model transitions. This puts the technical burden on the vendor — they must ensure that any new model meets the same quality standard as

the previous one before transitioning production workloads — while protecting the customer from quality degradation due to vendor infrastructure choices.

Data governance for AI service delivery is the fifth dimension. When the AI reviews the customer's contracts, it processes potentially highly confidential commercial information. The contract must specify: where is the processing performed (jurisdiction)? Is the data retained after processing? Does the vendor use customer data to improve the model? Can the customer audit the data handling practices? These questions are standard for enterprise SaaS but require more careful treatment for AI services because the nature of the data processing is more intimate — the AI is reading and analyzing content, not just storing and transmitting it.

Service as Software Contract — Five Key Dimensions

Dimension	What it covers	Standard approach	AI-specific complexity	Recommended clause design
SLA architecture	Performance commitments beyond availability: quality, throughput, turnaround	Standard SaaS: availability percentage only	AI quality is variable; throughput depends on input complexity; turnaround depends on queue depth	Specify quality metric (recall/precision); measurement methodology; independent auditor; credit formula; 2-tier remedy (credit then termination right)
Liability for AI errors	Who bears the cost of AI output errors that cause customer harm	Standard SaaS: liability capped at annual fees; no professional liability	AI provides professional-quality work product that customers rely upon; error can cause significant harm	Limit liability to annual fees + define reliance standard: customer may use AI output as input to professional judgment, not as substitute for it
IP ownership	Who owns the AI-generated work product	Standard SaaS: customer typically owns their data	AI output combines customer input data with vendor's model training —	Assign specific work products to customer; vendor retains model,

			ownership unclear without explicit assignment	prompts, methodology; address fine-tuning data ownership separately
Force majeure — model deprecation	What happens when the underlying AI model changes	Standard SaaS: force majeure covers infrastructure only	Foundation model deprecation can change service quality; vendor's choice to upgrade models is not traditional force majeure	Commit to quality standard regardless of model changes; define notice period (90 days) for major model changes; customer acceptance testing right before production switch
Data governance	How customer data is processed, retained, and used	Standard SaaS: data processing agreement under GDPR/CCPA	AI reads and analyzes sensitive content; model training on customer data creates additional complexity	Specify processing location; define retention (maximum 30 days for sensitive content); opt-out right for model training; deletion verification requirement

CONTRACTING INSIGHT

The SLA quality commitment is where Service as Software contracts most commonly fail — and where the most value is created

Most enterprise AI vendors avoid SLA quality commitments because they are hard to define and harder to measure. This avoidance is commercially expensive: customers who cannot get a quality commitment from Vendor A will pay a premium to Vendor B who provides one, even if Vendor A's actual quality is higher. Building the measurement infrastructure required to support a quality SLA is an investment in commercial differentiation, not just in product quality. The vendor who can say 'we guarantee 95% recall on material issues, measured by independent attorney review, with a defined credit formula for misses' is not just offering a quality guarantee — they are communicating commercial maturity that competitors without the infrastructure cannot match.

Chapter Nine — The Essentials

› Service as Software contracts require five AI-specific dimensions: SLA architecture, liability for errors, IP ownership, force majeure for model deprecation, and data governance.

› The quality SLA is the most commercially differentiating clause — vendors who support it pay the investment to build measurement infrastructure.

› Liability for AI errors: cap at annual fees plus define the reliance standard (AI supports professional judgment; does not substitute for it).

› IP ownership: assign work products to customer; vendor retains model, prompts, methodology — and address fine-tuning data separately.

› Force majeure for model deprecation: commit to quality standard regardless of model changes; 90-day notice for major model transitions.

CHAPTER TEN

Revenue Recognition for Service as Software

ASC 606 applied to outcome-priced AI services. When is a ticket 'resolved'? Who decides?

Revenue recognition for Service as Software contracts requires the Controller to navigate two genuinely novel accounting questions that the ASC 606 guidance does not address with specificity: when is a service "performed" for an AI-delivered service, and how is variable consideration constrained when performance depends on AI quality that is difficult to measure in advance?

The "when is it performed?" question arises most sharply in outcome-per-unit models. When Harvey AI reviews a contract and delivers a structured issues list, when is the performance obligation satisfied? Three options have been applied in practice:

Option one: at delivery. Revenue is recognized when the review document is delivered to the customer. This is the simplest approach and the most conservative — no risk of recognizing revenue before delivery. The accounting issue is that "delivery" may precede

the customer's acceptance, and the SLA may provide remedy rights (credits) that create variable consideration.

Option two: at delivery minus estimated credits. Revenue is recognized at delivery, net of an estimate of the credits that will be claimed based on historical SLA miss rates. This is technically correct under ASC 606 variable consideration treatment but requires a reliable estimate of the credit rate — which requires sufficient operating history with the specific SLA.

Option three: at acceptance. Revenue is recognized when the customer implicitly accepts the delivery (by taking the action that the review was meant to enable — signing the contract, for example) or explicitly accepts it (by confirming receipt without dispute within a defined window). This approach is most aligned with the principle that revenue should be recognized when control transfers to the customer with certainty, but it creates significant revenue timing unpredictability.

Most AI service companies have adopted option one or two, with the choice depending on the predictability of the SLA miss rate. A company with twelve months of data showing a 2% SLA miss rate can reliably estimate the credit reserve and apply option two. A company without that history should use option one to avoid the risk of overstating revenue.

The variable consideration constraint for quality-based pricing is the second novel accounting question. Consider a tiered outcome quality model where the customer pays $200 for Standard reviews and $450 for Professional reviews. If the Professional review does not meet its quality standard (recall rate below 95%), the customer receives a credit and the transaction price for that review is effectively the Standard price ($200). The revenue recognized for each Professional review is therefore variable: it is $450 if quality meets standard and $200 if it does not.

Under ASC 606, this variable consideration must be estimated and constrained. The constraint requires that it is highly probable that a significant revenue reversal will not occur when the uncertainty resolves. For a company with a 3% Professional-tier miss

rate (based on operating data), the expected revenue per Professional review is: (0.97 × $450) + (0.03 × $200) = $442.50. The constraint question is whether $442.50 is a reliable estimate that will not result in a significant reversal — which depends on whether the 3% miss rate is stable and whether it is measured with sufficient reliability.

The practical approach: apply the full $450 when the review is delivered and meets quality standard. Apply a credit when it misses. Track the miss rate monthly. If the miss rate exceeds the historical estimate, update the variable consideration estimate prospectively and reassess whether the constraint is still satisfied. Document the constraint analysis at each reporting date.

Multi-element arrangements in Service as Software contracts create additional allocation complexity. A contract that includes a platform subscription (access to the AI's interface), a quality assurance service (quarterly sample audits of AI output against expert human review), and consumption-based review credits is a three-element arrangement. The standalone selling price of the quality assurance service — which the vendor might not sell separately — must be estimated for the allocation. The Controller's methodology for this estimation must be documented and consistently applied.

Revenue Recognition Treatment by Service as Software Model

Model	Performance obligation satisfied when?	Variable consideration?	Constraint analysis	Key accounting risk
Outcome-per-unit	At verified outcome delivery — ticket closed without escalation; contract review delivered; invoice processed	Yes — SLA miss rate creates variable consideration (full price vs credit price)	Constrain if SLA miss rate > 5% or measurement unreliable; otherwise include full price	Recognizing revenue before verification complete; outcome definition insufficient to determine POC satisfaction
SLA-committed capacity	Ratably over the commitment period (access to committed capacity is the	Yes — SLA breach credits are variable consideration	Typically low constraint if historical SLA miss rate is stable; higher	Treating the subscription as fully earned regardless of capacity utilization — must assess if

	ongoing obligation)		constraint for new deployments	material underutilization is a performance failure
Tiered outcome quality	At delivery of work product at the committed tier quality standard	Yes — if quality misses tier standard, price reverts to lower tier (price concession)	Constraint depends on quality miss rate by tier; Expert tier typically lower miss rate than Standard	Recognizing Professional/Expert tier revenue before quality verification confirms tier standard was met
Gain-share	Base fee ratably; gain-share component at the agreed gain calculation date (annual or quarterly)	Yes — gain-share amount is entirely variable	Constrain gain-share component until sufficient operating history establishes reliable estimate; typically 12+ months before full recognition	Premature recognition of expected gain-share before the measurement period is complete and gain is verified
Subscription floor + overage	Floor: ratably · Overage: at unit delivery	Yes — overage volume is variable within the period	Low constraint on floor (committed); moderate constraint on overage if volume volatile	Allocating contract price between floor and overage components requires SSP estimation if not separately priced

Chapter Ten — The Essentials

› The question 'when is a service performed?' has a specific answer for each Service as Software model — document the determination before revenue is recognized.

› SLA breach credits in outcome-based models create variable consideration that requires estimation and constraint analysis.

› Gain-share components should be constrained until sufficient operating history establishes a reliable estimate — typically 12+ months of actual performance data.

› Tiered quality pricing creates a price-concession variable consideration: the effective price is the tier price if quality meets standard, the lower tier price if it does not.

› The Controller must document the performance obligation satisfaction criteria before the first invoice is issued — not as a post-hoc interpretation.

CHAPTER ELEVEN

The Operational Layer: Delivery, Quality, and SLA Management

Operationalizing outcome guarantees. Quality measurement systems. SLA breach protocols.

The operational layer of Service as Software — the systems and processes that govern delivery, measure quality, and manage SLA compliance — is where the commercial model becomes real for the customer. The most carefully designed pricing structure and the most precisely drafted contract are worth nothing if the operational layer cannot deliver on the commitments they create.

The operational infrastructure for Service as Software has three components: the delivery system, the quality measurement system, and the SLA management system.

The delivery system is the infrastructure that ensures AI work is delivered at the committed quality and turnaround time, at scale, without disruption. For an AI contract review service, this means: the AI processing pipeline with sufficient capacity to handle the committed volume plus peak buffers, the output formatting system that structures the AI's analysis into the defined deliverable format, the delivery mechanism that gets the finished review document to the customer through the agreed channel, and the delivery confirmation system that records each delivery with timestamp for SLA measurement.

The quality measurement system continuously monitors the AI's output against the quality commitments in the SLA. This is the most technically demanding component because it requires comparing AI output to a ground truth that is not always available in real time. Three approaches are used in practice:

Automated quality metrics measure the statistical properties of the AI's output — the length and structure of reviews, the distribution of issue severities, the consistency of language across reviews of similar documents. These metrics can be computed in real

time and can detect quality degradation (an AI that starts producing shorter, less detailed reviews may be experiencing model degradation) even before customer complaints arrive. They are not a substitute for outcome quality measurement but are an important early warning system.

Customer satisfaction sampling collects structured feedback from customers on a sample of deliveries. For a contract review service, this might involve a monthly survey of five randomly selected customers asking them to assess the quality of one recent AI review against their own judgment. The survey data provides direct customer quality feedback that complements the automated metrics.

Expert audit sampling uses human experts to independently review a sample of AI output against the quality standard in the SLA. This is the most credible measurement method for the SLA because it uses the same standard that the SLA references, but it is expensive and can only be applied to a sample. Most AI service companies conduct expert audits monthly or quarterly, with the sample size determined by the SLA credit threshold and the acceptable confidence level.

The SLA management system tracks performance against each SLA commitment in real time and generates the credit calculations when SLA misses occur. The system must be designed to prevent two common failure modes: late detection (discovering an SLA miss days after it occurred, when the customer has already noticed and escalated) and dispute generation (customers and vendors disagreeing about whether an SLA was missed because the measurement data is not shared).

The best SLA management systems are customer-facing: the customer has real-time access to the same SLA performance data that the vendor uses to calculate credits. When both parties are looking at the same numbers, disputes are rare. When the customer can only see the credit on their invoice without understanding how it was calculated, disputes are common.

Quality Measurement Approaches — Three Methods

Method	How it works	Cost profile	Reliability	Best used for
Automated quality metrics	Statistical properties of AI output computed in real time — review length, issue distribution, language consistency, response coherence	Very low — automated, no marginal cost per unit	Detects quality degradation trends; does not directly measure accuracy against ground truth	Continuous monitoring and early warning; supplement to other methods
Customer satisfaction sampling	Structured feedback from customers on sampled deliveries — monthly survey of 5 random customers on one recent AI output each	Low — survey administration + analysis cost	Customer perception of quality; biased by customer expertise and expectations	Customer relationship management and NPS-style quality tracking; inputs to CSM conversations
Expert audit sampling	Human experts independently review sample AI output against the SLA quality standard	High — requires domain expert time at market rates	Most credible for SLA compliance; uses same standard as the SLA	Quarterly SLA compliance verification; dispute evidence; annual audit reporting

OPERATIONS INSIGHT

Customer-facing SLA dashboards prevent disputes more effectively than any contract clause

The most common Service as Software dispute pattern: the customer believes the AI missed an SLA; the vendor believes it did not; neither has easy access to the shared performance data that would resolve the question. The solution is to make the SLA performance data customer-facing in real time. A customer who can log into a portal and see the AI's recall rate for the last 30 days, the average turnaround time for their submissions, and the credits they have accrued for SLA misses is a customer who does not need to dispute — they can verify. Build the customer-facing SLA dashboard before the first SLA dispute, not in response to it.

Chapter Eleven — The Essentials

› Quality measurement has three layers: automated metrics (continuous, low cost), customer satisfaction (relationship signal), expert audit (SLA compliance evidence).

› Expert audits should be conducted monthly or quarterly; the sample size is determined by the SLA credit threshold and required confidence level.

› SLA breach protocol: automated detection → immediate customer notification → credit calculation and application → root cause investigation → remediation plan.

› Customer-facing SLA dashboards prevent disputes: when both parties see the same data in real time, the basis for disputes disappears.

› The delivery confirmation system is the audit trail for SLA compliance — log every delivery with timestamp, and make the log queryable.

CHAPTER TWELVE

Competing with Your Own Customers: The Build vs Buy Dynamic

When customers can build their own service-as-software using the same models you use. The three strategic moats.

The build versus buy dynamic in Service as Software deserves extended treatment because it is the primary competitive risk that every AI service vendor faces — and the one that most commercial strategies need to specifically address.

The dynamic works as follows: the AI service vendor builds a product using foundation models from OpenAI, Anthropic, or Google. The product creates significant value for customers. The customers notice that the vendor's product is built on foundation models that the customers could, in principle, also access directly. The customers begin to evaluate whether they could build the AI service internally, using the same foundation models, and capture the value without paying the vendor's margin.

This evaluation is not hypothetical. It is happening right now, at every major enterprise AI buyer. Legal departments are evaluating whether to license Harvey or build a legal AI using Claude. Customer service operations are evaluating whether to use Intercom's AI or build an AI support agent using GPT-4o. Development organizations are evaluating whether to use Cursor or build an internal AI coding assistant.

The build versus buy decision is not always wrong for the customer. For organizations with strong AI engineering capabilities and clear strategic value in owning the AI infrastructure, building can be the right choice. But for most enterprises, it is not — the engineering investment, the ongoing model maintenance, the quality assurance infrastructure, and the domain expertise required to match a specialist vendor's performance are substantial.

The vendor's commercial strategy must address the build versus buy question directly, not hope it goes away. Three strategies have proven effective.

The first is the depth moat: investing in domain-specific capability that would take the customer years to replicate. Harvey's legal domain expertise — its fine-tuning on legal documents, its understanding of legal risk hierarchies, its integration with legal workflow systems — took Harvey three years to develop. A corporate legal department that decided to build internally would need comparable time and expertise. The depth moat is defensible because it is expensive to replicate.

The second is the workflow integration moat: embedding so deeply in the customer's operational workflow that switching to an internally built alternative would require redesigning core operational processes. An AI contract review system that is integrated with the customer's contract lifecycle management system, that automatically populates the review into the CLM workflow, and that tracks issue resolution through the negotiation process is not just a review tool — it is an operational infrastructure component. The switching cost includes not just replacing the AI but rebuilding all the integrations.

The third is the accountability moat: the vendor accepts accountability for the quality of the AI's work in a way that an internal team cannot easily replicate. When Harvey provides contract reviews with a defined quality guarantee and a credit mechanism for misses, it is accepting professional accountability for the work. An internal team building the same capability would need to establish comparable quality standards, measurement infrastructure, and accountability structures — all of which require time,

investment, and organizational commitment that most enterprises are unwilling to make for a supporting function like contract review.

The commercial design of Service as Software products should deliberately build these moats into the product architecture, not just the feature set. The quality guarantee, the SLA structure, the workflow integration, and the domain expertise are not just commercial features — they are the competitive infrastructure that makes the build versus buy calculation favor the vendor.

Build vs Buy Decision Factors — Vendor's Competitive Analysis

Factor	Build case	Buy case	Moat implication
AI engineering capability	Strong AI/ML team; strategic intent to own AI infrastructure; ability to maintain models	Limited AI expertise; AI is supporting function not strategic core; maintenance burden unacceptable	Vendors serving buyers with strong engineering teams need a stronger moat — depth of domain expertise matters more
Time to value	Can build comparable capability in 3–6 months	Building equivalent capability would take 18–36 months including training data, fine-tuning, quality infrastructure	The time moat is real but temporary — vendors who do not add deeper moats will face build competition as AI tools improve
Compliance and accountability	Internal team can meet compliance requirements; no need for external accountability	Need external vendor's compliance certification; accountability for AI errors requires vendor structure	Accountability moat is most durable — enterprises will continue to need external accountability for professional-quality AI work
Ongoing improvement	Can maintain and improve internal AI over time	Vendor's continuous improvement provides better AI without internal investment	Improvement moat: vendors who are improving faster than customers can build will always be ahead; commoditization narrows this advantage
Total cost of ownership	Cost of building = one-time (if successful)	Cost of buying = ongoing; may be lower than building plus maintaining	TCO is complex for AI — building is often underestimated; vendors who can demonstrate TCO

			advantage win the build vs buy conversation

Chapter Twelve — The Essentials

› The build vs buy evaluation is happening at every major enterprise AI buyer right now — vendors must address it in their commercial strategy, not hope it goes away.

› Three durable moats: depth (domain expertise too expensive to replicate), workflow integration (switching requires re-engineering operations), accountability (vendor accepts professional accountability).

› The accountability moat is most durable: enterprises cannot replicate external accountability structures for professional-quality AI work.

› Time-to-value is a real moat but not a permanent one — it buys time; it does not replace a deeper competitive advantage.

› The commercial design should deliberately build moats: quality guarantees (accountability), workflow integration points (switching cost), domain expertise depth (replication barrier).

PART FOUR

Strategy and the Road Ahead

Margin architecture. Customer success redesign. Enterprise governance. The category's future.

CHAPTER THIRTEEN

Defending Margin When You Are Selling Services at Software Prices

Gross margin architecture for service-as-software. Where costs live. How to protect them.

Defending gross margin in Service as Software requires understanding precisely where margin is created and where it is at risk — which is different from the margin analysis of traditional software or traditional services.

Traditional software has a simple gross margin structure: revenue is the software license or subscription fee; COGS is the infrastructure cost to host and operate the software (typically 10–25% of revenue for mature SaaS companies). Gross margins above 70% are standard and above 80% are achievable. The margin is high because the variable cost of adding another user is near zero.

Traditional services have the opposite structure: revenue is the billing rate; COGS is the labor cost (typically 40–60% of revenue for professional services, higher for commoditized services). Gross margins of 25–40% are typical for professional services firms. The margin is lower because the variable cost of delivering another unit of service — an additional hour of professional labor — is substantial.

Service as Software exists between these two poles, but closer to the software side in economics. The variable cost of delivering another unit of AI service (an additional contract review, an additional support ticket resolution, an additional invoice processed) is the marginal model inference cost — which for current models is typically $5–20 per complex professional work product. At a price of $150–600 per unit of professional work, the gross margin on the variable component can be 90–97%.

However, Service as Software has cost categories that pure software does not: quality assurance infrastructure (the expert audits and measurement systems required to maintain the quality guarantee), SLA management infrastructure (the systems that track performance and calculate credits), professional liability insurance (required when the AI's output is relied upon for professional purposes), and the expected cost of SLA credits (which must be provisioned even if not paid in every period).

The gross margin waterfall for a contract review AI service at scale looks approximately like this: Revenue per review: $300. Model inference cost: -$12. Quality assurance per

review (amortized audit cost): -$8. SLA credit reserve (3% miss rate × credit amount): -$7. Infrastructure and delivery: -$5. Gross margin per review: $268, or 89%.

This margin profile is extraordinary — substantially better than professional services (40% gross margin for the comparable human service) and competitive with mature SaaS (80–85% gross margins). The margin advantage derives entirely from the AI's variable cost being near zero relative to the human labor cost of comparable service delivery.

The risk to this margin profile comes from three directions. First, model cost inflation: as AI models become more capable, they become more expensive. If the market for contract review services becomes competitive enough that vendors cannot raise prices to offset model cost increases, margin will compress. Second, quality assurance cost escalation: as customers become more sophisticated about AI quality, the evidentiary standard for quality measurement will increase, potentially requiring more frequent and more expensive expert audits. Third, pricing pressure from commoditization: if the foundation models underlying Service as Software products become sufficiently capable that the differentiation between products narrows, price competition will erode the premium that quality differentiation currently supports.

The margin defence strategy has three components that mirror the three risks. Against model cost inflation: invest in prompt optimization, caching, and model selection to reduce the inference cost per delivered work product. A contract review AI that uses GPT-4o for initial classification and Claude 3.5 Sonnet for deep analysis, with caching for repeated document patterns, might achieve the same output quality at 40% lower inference cost than using the same frontier model for every step. Against quality assurance cost escalation: invest in automated quality metrics that reduce the reliance on expensive expert audits. If the automated metrics can reliably predict the outcomes of expert audits, the audit frequency can be reduced without sacrificing the quality guarantee's credibility. Against pricing pressure: invest in the depth moat and workflow integration moat described earlier. These investments are margin-defending investments as much as competitive investments.

Service as Software Gross Margin Waterfall			
Line item	**Illustrative amount (per contract review)**	**% of revenue**	**Notes**
Revenue per unit	$300	100%	Per-review price for Professional tier contract review
Model inference cost	-$12	-4%	Direct model API cost at current market rates; subject to model cost changes
Quality assurance (amortized)	-$8	-2.7%	Expert audit cost amortized across all reviews in the audit sample
SLA credit reserve	-$7	-2.3%	3% miss rate × ($300 - $150 Standard price equivalent) × probability-weighted reserve
Infrastructure and delivery	-$5	-1.7%	Delivery infrastructure, output formatting, delivery confirmation system
Gross profit per unit	$268	89.3%	Extraordinary margin profile — better than SaaS, incomparably better than human services
For comparison: human paralegal	$600	Revenue; $360 direct cost = 40% gross margin	The AI delivers comparable quality at half the price to the customer, with 89% margin vs 40%

Three Margin Risks and Defence Strategies				
Risk	**Mechanism**	**Magnitude**	**Defence strategy**	**Investment required**
Model cost inflation	Foundation model prices increase as models become more capable; vendor cannot pass full increase	Moderate; model prices have trended down historically but capability jumps can	Prompt optimization and model routing: use cheaper models for sub-tasks where quality is adequate; frontier model only for	6–12 months of ML engineering investment in prompt optimization and model selection framework

	through to customers	cause price increases	quality-critical steps. Target: 40% inference cost reduction from routing optimization.	
QA cost escalation	Customer expectations for quality evidence increase; expert audit frequency and scope requirements grow	Moderate but manageable; professional audit rates are relatively stable	Automated quality metrics that reduce reliance on expert audits: if automated metrics reliably predict expert audit outcomes, audit frequency can be reduced without sacrificing credibility. Target: 50% reduction in audit cost through automation.	3–6 months to build automated quality metrics that correlate strongly with expert audit results
Pricing pressure from commoditization	Competing AI services offer comparable standard-tier quality at lower prices; premium compression	High risk for Standard tier; lower for Expert tier (accountability moat protects premium)	Invest in domain depth (harder to replicate) and workflow integration (switching cost). Migrate revenue toward Expert tier where the accountability moat is strongest. Price Standard tier competitively; defend Expert tier margin.	Ongoing investment in domain expertise and integration depth; managed as a portfolio

Chapter Thirteen — The Essentials

› The gross margin profile for Service as Software is extraordinary: 89–96% on the variable component when AI replaces human professional labor at comparable quality.

› The margin waterfall has four costs: model inference, QA amortized, SLA credit reserve, and delivery infrastructure.

› Three margin risks: model cost inflation (defend with routing optimization), QA cost escalation (defend with automated metrics), commoditization (defend with accountability moat).

› The Expert tier with accountability is the most durable margin source — competitors cannot undercut it without accepting comparable accountability.

› Model routing optimization is the highest-leverage near-term margin investment: using cheaper models for quality-adequate sub-tasks reduces inference cost 30–50% without quality impact.

CHAPTER FOURTEEN

The Customer Success Model for Service as Software

Outcome verification, SLA management, and expansion in a product that runs autonomously.

Customer success for Service as Software products operates on a fundamentally different logic from customer success for traditional SaaS. The difference is what "success" means.

In SaaS, customer success is measured by adoption: are users using the product? Are they using the right features? Is the product embedded in their workflows deeply enough that switching costs are high? The customer success motion is adoption-focused: onboard users effectively, drive feature adoption, identify expansion opportunities, manage renewal risk based on usage data.

In Service as Software, "success" is not about adoption. The AI uses itself. A legal department that has deployed Harvey for contract review does not "adopt" Harvey in the way they adopt CRM software — the lawyers do not log into Harvey. The AI reviews contracts on their behalf. Success is not measured by adoption; it is measured by outcomes.

This requires a completely different set of metrics for the customer success function.

Outcome delivery rate is the primary health metric: what percentage of submitted work products met the defined quality standard? A contract review AI with a 97% outcome delivery rate is delivering on its commitments; one with an 82% delivery rate is experiencing systematic quality issues that will generate credits, customer dissatisfaction, and renewal risk.

Time-to-value acceleration measures how quickly the AI's output is integrated into the customer's decision workflow. A contract review AI is only delivering value if the reviewed contracts are actually advancing through the legal department's approval process, not sitting in a reviewed-but-unread queue. Time-to-value metrics require the customer success team to understand the customer's downstream workflow, not just the upstream product usage.

Outcome verification tracking follows the verified outcomes through to their business impact. A customer service AI that resolves 90% of tickets without escalation is delivering on its SLA. But if those resolved tickets are generating higher-than-normal re-contact rates (customers who were "resolved" but came back with the same issue), the outcome quality is lower than the headline metric suggests. CSMs for Service as Software products need access to downstream outcome data, not just the AI's self-reported performance.

Expansion signal identification in Service as Software is driven by outcome data, not usage data. A law firm that is using Harvey for NDA review and whose Harvey-reviewed NDAs are completing the approval process 40% faster than non-Harvey-reviewed agreements has data that supports a conversation about expanding Harvey's scope to more complex agreement types. The expansion signal is in the outcome comparison, not in the feature utilization report.

The CSM's role in Service as Software is therefore more analytical and more commercially oriented than in SaaS. Instead of "are users logging in?", the CSM asks "what is the AI accomplishing, and is the accomplishment valuable enough to justify the contract at renewal and expansion?" This requires the CSM to have access to financial data (what is the dollar value of the AI's contribution to this customer's operations?), to

process data (how has the customer's operational workflow changed since AI deployment?), and to competitive data (how does this customer's AI performance compare to comparable customers?).

The compensation structure for Service as Software CSMs should reflect this shift. SaaS CSMs are often compensated primarily on renewal rate and net revenue retention — metrics that are substantially driven by adoption and health score. Service as Software CSMs should be compensated on outcome delivery rate, time-to-value metrics, and expansion revenue — metrics that are driven by the AI's performance and the CSM's ability to convert outcome evidence into commercial expansion.

Customer Success KPIs: SaaS vs Service as Software

KPI	SaaS customer success	Service as Software customer success	Why it changes
Primary health signal	Login frequency; feature adoption; active users	Outcome delivery rate; SLA compliance; time-to-value acceleration	The AI uses the product; human adoption metrics measure the wrong thing
Expansion trigger	New use case identified; team growth; feature request	Outcome data shows AI deployment depth below potential; performance comparison to similar customers	Expansion is data-driven, not sales-driven
Renewal evidence	Adoption metrics; customer satisfaction scores	Outcome ROI calculation; dollar value of AI contribution to customer operations	The renewal conversation is about demonstrated value, not future potential
Churn prediction	Low login frequency; support ticket increase; disengagement signals	Outcome quality decline; AI performance below SLA; customer submitting fewer work items to AI	AI performance data predicts churn earlier than engagement data
CSM activity	Feature training; business reviews; use case discovery	Outcome reporting; performance benchmarking; expansion ROI modeling	CSM role shifts from educator to analyst and commercial advocate
Compensation driver	Renewal rate; NPS; QBR completion	Outcome delivery rate; expansion revenue;	CSM incentives must align with what Service as Software measures

		customer improvement	P&L

Chapter Fourteen — The Essentials

› Customer success for Service as Software is outcome-driven, not adoption-driven — the AI uses the product; humans do not need to be adopted.

› Primary health metric: outcome delivery rate (is the AI delivering on its commitments?), not login frequency.

› Expansion is driven by outcome data: customers whose AI deployment is underperforming relative to its potential are expansion conversations grounded in evidence.

› The renewal conversation is about demonstrated ROI: dollar value of AI contributions to customer operations, compared to the cost of the service.

› CSM compensation must align with outcome metrics — renewal rate and NPS do not capture what matters in this category.

CHAPTER FIFTEEN

Service as Software at Enterprise Scale: Governance and Trust

Enterprise procurement, security, audit, and governance requirements for service-as-software vendors.

Enterprise governance requirements for Service as Software vendors are substantially more demanding than for traditional SaaS vendors, because the AI is performing work that has professional, legal, and regulatory consequences — not just providing a tool that humans use to perform that work.

The distinction matters commercially. When Salesforce provides CRM software, the consequences of a software error fall primarily on the user who made a decision based on incorrect information. When Harvey AI provides contract review, the consequences of an AI error can include a contract with missed issues, financial exposure from the

missed issue, and potential professional liability for the legal team that relied on the review. The enterprise buyer's governance requirements reflect this difference in consequence.

Security and data handling requirements for Service as Software are more stringent than for SaaS. The AI is processing the customer's most sensitive documents — contracts, financial records, clinical notes, personnel files. The data handling requirements must address: where the processing occurs (data sovereignty requirements), how long data is retained (minimum viable retention to enable quality improvement, maximum retention to limit exposure), whether the data is used for model training (opt-out or opt-in mechanisms), and what happens to the data if the commercial relationship ends (deletion with verification).

Regulatory compliance requirements are specific to each sector. Legal AI deployed in a regulated industry (financial services, healthcare) must comply with the regulations applicable to the work product it generates — in some jurisdictions, the legal review of a contract by an AI system may require disclosure that AI was used. Healthcare AI that generates clinical notes must comply with CMS documentation requirements and HIPAA data handling standards. The vendor's compliance posture must match the regulatory requirements of the customer's sector.

Explainability requirements for AI decisions are emerging as a governance requirement in regulated industries. An AI that reviews a contract and identifies an issue must be able to explain why it identified the issue — not just what it found, but why the language it flagged represents the risk it claims. This explainability requirement is not currently universal but is increasingly demanded by enterprise risk management functions as AI is deployed for higher-stakes work. Service as Software vendors who invest in explainability infrastructure — who can provide, for any AI-generated finding, a clear explanation of the reasoning that produced it — are building a governance advantage that will become increasingly commercially relevant.

Human oversight mechanisms are required for the most consequential AI work. No enterprise will allow an AI to take consequential action — execute a contract, make a

lending decision, discharge a patient — without a human review step. The Service as Software commercial model must accommodate these oversight requirements: the pricing must reflect the oversight cost (if the vendor provides the human oversight, it is part of the service cost; if the customer provides it, the price should reflect that they are contributing labor to the service delivery), and the SLA must address what happens when human oversight creates a bottleneck (if the AI delivers the review but the human reviewer is unavailable, whose performance obligation has been missed?).

Audit and accountability requirements are the final governance dimension. Enterprise customers increasingly require the ability to audit the AI's decisions — to understand, for any specific output, what data was processed, what model was used, what reasoning was applied, and who or what had access to the output. This audit requirement is both a data security requirement (the customer needs to know what happened with their sensitive documents) and a quality assurance requirement (the customer needs to be able to investigate any output they believe was incorrect).

The Service as Software vendor that builds audit infrastructure — detailed logs of every AI action, explainable reasoning for every AI output, accessible audit trails for every data processing event — is building governance capability that enterprise customers will pay for. The vendor who treats audit as a compliance afterthought will find that their governance posture becomes a deal blocker as enterprise governance requirements increase.

Enterprise Governance Requirements — Service as Software

Requirement	What enterprise buyers need	Why AI services differ from SaaS	Vendor investment required
Security and data handling	Data processing in approved jurisdictions; minimum retention; opt-out from model training; deletion verification	AI processes the content of sensitive documents, not just their metadata — more intimate data processing requires more stringent handling	Security attestations (SOC 2 Type II); DPA templates for major jurisdictions; verifiable deletion process; customer-controlled data retention configuration

Regulatory compliance	Compliance with sector-specific regulations for AI-generated work products	AI work products may themselves be subject to regulation (CMS documentation standards, SOX-relevant outputs)	Sector-specific compliance posture; regulatory counsel relationships; compliance documentation for each regulated sector served
Explainability	Ability to explain specific AI findings and recommendations	Professional-quality AI work creates implicit accountability for the reasoning behind findings	Reasoning documentation for each output; audit trail of data accessed; model decision transparency at the finding level
Human oversight	Review and approval mechanisms for consequential AI actions	AI taking consequential action requires a human review gate at enterprise scale	Configurable human-in-loop workflow; escalation mechanisms; review queue management
Audit capability	Ability to audit AI decisions for compliance and quality assurance	Enterprise risk management requires ability to audit AI-generated work products	Complete audit trail: data processed, model version, reasoning applied, human review actions, output delivered and accepted

Chapter Fifteen — The Essentials

› Enterprise governance requirements for Service as Software are more demanding than for SaaS because the AI performs professional-quality work with professional-quality consequences.

› Security and data handling: process in approved jurisdictions, minimum content retention, verifiable deletion, opt-out from model training.

› Explainability: vendors who invest in reasoning documentation for AI findings are building a governance moat that compliance-sensitive enterprises will pay for.

› Human oversight mechanisms are non-negotiable for consequential AI actions — design the commercial model to accommodate oversight requirements.

› The vendors who build enterprise governance infrastructure proactively (before it is demanded) will close deals that those who retrofit governance reactively will lose.

CLOSING

The Destination — and the Map

Service as Software is not a variant of SaaS. Design for it deliberately.

Framework F19 — The Service-Software Convergence Model

Framework F19 — The Service-Software Convergence Model — can be summarized as follows.

The convergence model identifies five positions on the tool-to-service spectrum: pure tool (human fully operates the software), AI-assisted tool (AI augments human operation), AI-augmented workflow (AI does most of the work, human reviews and approves), autonomous workflow (AI executes complete workflows, human monitors), and pure AI service (AI delivers professional-quality work product directly).

For each position on the spectrum, the model specifies: the appropriate billing unit (per access, per consumption, per workflow, per outcome), the natural pricing model (subscription, hybrid, per-task, per-outcome), the SLA commitment that is appropriate (availability, quality, turnaround), the liability model (software limitation, shared responsibility, professional accountability), and the customer success motion (adoption-driven, outcome-driven, performance-driven).

The model's central finding is that most AI software companies are operating at positions 3 or 4 on the spectrum (AI-augmented workflow or autonomous workflow) while pricing at position 1 or 2 (subscription access or consumption usage). This misalignment is the open-claw gap expressed in commercial architecture terms: the product is delivering service-level value; the pricing is capturing tool-level value.

The convergence model is a migration roadmap as much as a diagnostic. A company that identifies its current position on the spectrum can trace the path from current pricing to appropriate pricing, with the operational prerequisites, the customer communication requirements, and the contractual changes required at each step.

The companies that navigate this migration successfully will look, in five years, more like professional service firms than like SaaS companies — not in their cost structure (their economics will remain software-like, with high gross margins), but in their commercial relationship with customers. They will charge for what they deliver, not for access to the capability that delivers it. They will be accountable for quality, not just for availability. They will grow their revenue with each customer's AI deployment depth, not with their headcount.

That is the destination. This book is the map.

Framework F19 — The Service-Software Convergence Model

Spectrum position	Billing unit	Natural pricing model	SLA commitment	Customer success motion	Commercial moat
1. Pure tool	Per user / per seat	Subscription	Availability (uptime)	Adoption-driven	Feature richness; switching cost
2. AI-assisted tool	Per user + consumption	Subscription + consumption add-on	Availability + response time	Adoption + feature utilization	AI quality; integration depth
3. AI-augmented workflow	Per workflow or hybrid	Hybrid subscription + per-task	Availability + quality (partial)	Outcome-driven begins	Workflow integration; outcome measurement
4. Autonomous workflow	Per task / per outcome	Per-task + subscription floor	Availability + quality + throughput	Outcome-driven	Domain depth; accountability; switching cost
5. Pure AI service	Per work product	Per-outcome + gain-share option	Quality + throughput + accuracy	Performance-driven	Professional accountability; domain expertise; audit infrastructure

The convergence between software and services is not a trend. It is an irreversible change in what software products are. When an AI agent does the work that a human

used to do — reviews the contract, resolves the ticket, writes the code, processes the invoice — the nature of what is being sold changes fundamentally. The customer is no longer buying a tool. They are buying the work.

The commercial implications follow from this change with the logic of physics. If you are selling the work, you should price the work. Not the tool used to do it. Not the resources consumed to do it. Not the time spent on it. The work itself — the verified outcome, the delivered service, the accomplished task.

This is both the simplest and the most demanding proposition in AI monetization. Simple because the logic is clear. Demanding because the operational infrastructure required to make it real — the outcome definition, the measurement system, the quality guarantee, the contractual accountability, the governance infrastructure — is genuinely complex.

The companies that build that infrastructure will own the AI economy's most valuable commercial positions. They will have pricing power that commodity AI cannot touch, because they are charging for results, not for access. They will have customer relationships built on accountability, not just on feature richness. They will have expansion mechanics that grow automatically with AI performance improvement, without requiring the perpetual renegotiation that seat-based models demand.

The companies that do not build it will find themselves in an increasingly uncomfortable position: charging for access to tools that customers know are doing professional-quality work, watching competitors who charge for the work itself capture the value that the access-based model leaves on the table.

The question is not whether AI has changed what you sell. It has. The question is whether your pricing model knows it yet.

> ***"The question is not whether AI has changed what you sell. It has. The question is whether your pricing model knows it yet."***

The AI Economy Monetization Series continues in Book Six:

When Software Is a Commodity: How to Monetize and Win